IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPELLANT'S MAIN BRIEF ON APPEAL

APPLICANT:	Karlheinz Dorn, et al.	DOCKET NO:	P02,0630-01
SERIAL NO.:	10/725,110	ART UNIT:	2136
FILED:	December 1, 2003	EXAMINER:	Louie, Oscar A.
CONF. NO.:	3328		

TITLE: Procedure for User Login to Data Processing Devices

Mail Stop Appeal Brief-Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. §41.37, Appellant submits this Brief in support of the appeal of the above-referenced application in support of the patentability of claims 1–11 finally rejected in the Office Action, dated May 29, 2007. Applicants filed Amendment B After Final on July 30, 2007, and the Examiner entered the amendment, but indicated that the claims remained unallowable in an Advisory Action mailed August 9, 2007. A copy of the claims on appeal is attached as Appendix A. A Notice of Appeal was filed on August 28, 2007.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the assignee, Siemens Aktiengesellschaft, a German corporation, by virtue of the Assignment recorded April 12, 2004, at reel/frame 015199 / 0262.

RELATED APPEALS AND INTERFERENCES

There are no related appeals and no related interferences known to Appellant, Appellant's Assignee, or Appellant's legal representative.

STATUS OF CLAIMS

Claims 1-11 are on appeal, and constitute all pending claims of the application. In the Final Office Action (FOA), in ¶1, the claims were rejected as follows:

Claims / Section	35 U.S.C. Sec.	References / Notes
10	§101 Non-statutory subject matter	• Claim is directed to a computer program. Overcome by Amendment after Final
1–5 & 7–11	§102(b) Anticipation	• Dutcher (U.S. Patent No. 6,021,496).
6	§103(a) Obviousness	• Dutcher (U.S. Patent No. 6,021,496); and • Win (U.S. Patent No. 6,161,139).

STATUS OF AMENDMENTS

Amendment B After Final was filed on July 30, 2007, following the final rejection, that amended claim 10 to overcome the 35 U.S.C. §101 rejection, but that argued the remaining bases for rejection.

An Advisory Action (AA) was mailed on August 9, 2007, and indicated that this amendment was entered and did overcome the rejection under 35 U.S.C. §101, but otherwise did not place the application in a condition for allowance.

SUMMARY OF THE CLAIMED SUBJECT MATTER

The use of page and line numbers and reference characters in the drawings in the following summary is provided by way of example and is in no way intended to limit the claimed subject matter unless expressly indicated.

The following summary of the independent claims is provided pursuant to 37 C.F.R. §41.37. Paragraph numbers are referenced in the form [xxxx], otherwise, number indicate reference characters or Figure numbers.

Claim 1. A method for logging (Fig. 2, 1 [0016]) a new user into a data processing device with an operating system (79) and an accessible element (71, 73, 85) that is at least one of an application program (71, 73) and sensitive data (85), comprising the steps of:

ending a first user's access to the accessible element (71, 73, 85) without unloading or restarting the accessible element (Fig. 2, 37, 39, [0058. 0059]);

determining authentication data for authenticating a second user (Fig. 1, 77, Fig. 2 7–11, [0049]);

defining an identity and access rights depending on the authentication data for the second user; (Fig. 2 13) and

providing access, depending on the defined access rights, for at the accessible element, that has not been unloaded or restarted, by the second user. (Fig. 2, 15, 35, 37, [0051–0054]).

Claim 10. A computer system [0085] comprising:

a computer having a memory [0085];

an accessible element (Fig. 1 71, 73, 85) that is at least one of an application program (71, 73) and sensitive data (85) that is accessible by a first user and a subsequent second user without unloading or restarting the accessible element [0058];

a program stored in a memory element of the computer memory comprising [0085]:

a software module or algorithm [0085] for determining authentication data for authenticating the second user with respect to the accessible element (Fig. 1 77, Fig. 2 7–11, [0049]) ;

a software module or algorithm [0085] for defining an identity and access rights depending on the authentication data (Fig. 2 13, [0050]); and

a software module or algorithm [0085] for providing access, depending on the defined access rights, for the accessible element (Fig. 2 15, 35, 37, [0051–0054, 0058]).

Claim 11. A computer readable data storage media having a program thereon [0026] that comprises:

a software module or algorithm [0085] for determining authentication data for authenticating a user into a data processing device (Fig. 1 77, Fig. 2 7–11, [0049]) with an operating system (79) and an accessible element (71, 73, 85) that is at least one of an application program (71, 73) and sensitive data (85);

a software module or algorithm [0085] for defining an identity and access rights depending on the authentication data (Fig. 2 13, [0050]); and

a software module or algorithm [0085] for providing access by the user, depending on the defined access rights, for the accessible element subsequent to an access of the accessible element (Fig. 2 15, 35, 37, [0051–0054, 0058]) by a prior first user without unloading or restarting the accessible element ([0058]).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The issue on appeal is whether the subject matter of claims 1–5 and 7–11 are anticipated under 35 U.S.C. §102 by Dutcher (U.S. Patent No. 6,021,496). Claim 6 is not argued separately.

ARGUMENT

ARGUMENT –Anticipation of Claims 1–5 and 7–11 in View of Dutcher

Examiner's Position: Dutcher anticipates 1–5 and 7–11 because it teaches each and every element of these claims. Dutcher's disclosure of a networked Windows NT client-server environment implicitly teaches multiple user access to an accessible element.

In the FOA, on pp. 2–5, the Examiner rejected independent claim 1 as being anticipated by Dutcher and indicated how Dutcher was being read on each of the elements of claim 1.

With regard to the last claimed element, the Examiner stated, on pp. 4–5:

> providing access, depending on the defined access rights, for at least one of the application program and

> sensitive data" (i.e. "Thus, according to a primary goal of the present invention, the homogeneous NT client-server environment is uncoupled so that a user of a Windows NT client (by way of example only) may be authenticated by a nonnative server. With respect to authentication of the Windows NT client, the client-server environment is "heterogeneous." Authentication at the client gives the user access to resources on the client system, and when this is done via an account definition held at a server, it also gives the user access to resources at the server network via a single logon.
>
> The present invention thus enables a user to select a particular location against which he or she desires to be authenticated. Thus, the user's account information may be retained at the non-native server domain in addition to (or instead of) the Windows NT server normally coupled to the Windows NT client in a closed manner. The user's single userid and password are then held out at a non-native server, such as a Warp Server, a DCE cell, or the like. This information may also be retained at a native server domain") [column 6 lines 1-18];

In Amendment B After Final, the Appellants noted that the language used in the FOA by the Examiner related to the language of claim 1 before it had been changed by Amendment A, and that this amendment was significant. The previous Amendment A changed the language of this element of claim 1 to read from:

> “providing access, depending on the defined access rights, for at least one of the application program and sensitive data”

to

> “providing access, depending on the defined access rights, for the accessible element *that has not been unloaded or restarted, by the second user*”.

The Examiner then replied in the Advisory Action (AA) in three paragraphs:

First Paragraph

- Applicant's arguments regarding Claims 1-5 & 7-11, state that the language now reads, "providing access, depending on the defined access rights, for the accessible element that has not been unloaded or restarted, by the second user."

The applicant further argues that "it requires that the access be performed without an unloading and reloading" and "a preceding (or 'first') user and a loading/restarting (or lack thereof, as required by the claims), with a subsequent access by a second user."

Dutcher discloses a native/non-native server domain using Windows NT [column 6] as being one possible client server configuration. This implies that there would be multiple users (i.e. a first user, a second user, a third user etc) with individual user accounts whose activities do not disrupt one another as would be the purpose of having a domain server.

This also implies that applications do not need to be restarted for one user simply because another user logs off/on, executes applications, or performs other tasks and functions, since the clients have individual user accounts on a domain with multiple shared resources, as is the nature of a Windows NT client server operation.

Dutcher may not explicitly use the same language as the applicant in regards to "a first" and "second user" but Dutcher's disclosure encompasses multiple users who would use the client server system at varying times in relation to one another. The examiner notes that "a first" and "second user" would be implied in a client user system (i.e. Windows NT) on a network domain.

Second Paragraph

- Applicant's arguments recited as, "the system of Dutcher would naturally deal with more than one user, it does not contain any disclosure with respect to the access of applications or application data by multiple users in the context of access authorizations, and contains no disclosure with regard to the maintaining of an application or respective data in memory after use by the first user for use by the second user."

Dutcher discloses, "The invention allows authentication out to any number of non-native servers, as well as authentication from NT

> domains...The NT user account allows the authenticated user to obtain the server's resources as well as local resources ...One of the preferred implementations of the invention is a client application, namely, a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory" [columns 12-13].
>
> This disclosure teaches that Dutcher does disclose coverage for authentication and control over data/resources relating to the authentication, as well as, maintaining application data in memory until the computer requires/is instructed otherwise.

Third Paragraph

> - Applicant's argument regarding independent Claim 1 additionally states, "Dutcher does not address such a subsequent accessing by a first and second user."
>
> Dutcher discloses "a user of a Windows NT client" [column 6] which implies that there would be one of many users, where each user has access to their own Windows NT client as a part of a domain on a network.
>
> It is noted that any computer system comprising user logon and authentication on a network in a client server configuration comprises at least two or more users, hence the definition of a computer network.

Appellants' Position: Dutcher fails to teach or suggest at least the last element of independent claim 1 (and similar elements of independent claims 10 and 11) in which access is provided to the accessible element accessed by a first user, that has not been unloaded or restarted, by a second user depending on defined access rights of the second user.

An example of an embodiment of the present invention is provided in the Specification in which, in a medical environment (e.g., a medical imaging context) dealing with large amounts of private data (e.g., a diagnostic image of a patient), a second user can access such data according to his level of authentication without the application data being discarded or an application being terminated when a first user is done using the application/data. This permits a rapid change over from one user to a second user in a context where speed can be vitally important (see Summary section of the Specification).

Dutcher does not address such a subsequent accessing by a first and second user. Dutcher deals with the general issue of authorization and access in a networked environment and the specific issue of obtaining (for a particular user) access authorization where the access authorization may be native (e.g., Windows-based in a Windows environment) or non-native (UNIX-based in a Windows environment). It deals with the allocation of processes and procedures associated with a user's logon to a system within a domain.

Dutcher deals with the situation of a single user and multiple authentication providers (domain drivers) (14:26–28). Although the system of Dutcher would naturally deal with more than one user, it does not contain any disclosure with respect to the access of applications or application data by multiple users in the context of access authorizations, and contains no disclosure with regard to the maintaining of an application or respective data in memory after use by the first user for use by the second user.

This distinction found in the present invention is not just an obvious variant of what is disclosed in Dutcher, since the present invention advantageously permits much greater speed and efficiency for multiple users with potentially differing levels of authority and access privileges to access the large volumes of data and applications that deal with them typically found within the medical community—a problem that the system of Dutcher fails to address.

In the Examiner's AA, first paragraph, he states

> Dutcher discloses a native/non-native server domain using Windows NT [column 6] as being one possible client server configuration. This implies that there would be multiple users (i.e. a first user, a second user, a third user etc) with individual user accounts whose activities do not disrupt one another as would be the purpose of having a domain server.

Appellants do not disagree with what the Examiner finds to be implied here. Certainly in a Windows NT client-server environment, one would expect to find the possibility of multiple users each having individual user accounts. The

Examiner states:

> This also implies that applications do not need to be restarted for one user simply because another user logs off/on, executes applications, or performs other tasks and functions, since the clients have individual user accounts on a domain with multiple shared resources, as is the nature of a Windows NT client server operation.

Appellants do not agree with what the Examiner finds to be implied here. Whether or not an application or its associated data is persistent cannot automatically be inferred from the teaching of Dutcher. In such architectures, generally, if an application is not being used by someone, it is unloaded in order to free up resources for other applications. Often separate instances of an application may be created, and each with a user's own private space for the temporary storage of data. In any case, the Examiner has failed to point to an element identified in Dutcher as the claimed "accessible element" and shown how it is accessed by a first and second user without an unloading or restarting of the accessible element between access by the first and second user.

The Examiner then goes on to state:

> Dutcher may not explicitly use the same language as the applicant in regards to "a first" and "second user" but Dutcher's disclosure encompasses multiple users who would use the client server system at varying times in relation to one another. The examiner notes that "a first" and "second user" would be implied in a client user system (i.e. Windows NT) on a network domain.

Again, Appellants do not disagree with this characterization—the use of a system such as that disclosed by Dutcher would certainly imply a use by more than a single user.

In the AA second paragraph, the Examiner states:

> Dutcher discloses, "The invention allows authentication out to any number of non-native servers, as well as authentication from NT domains...The NT user account allows the authenticated user to obtain the server's resources as

> well as local resources ...One of the preferred implementations of the invention is a client application, namely, a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory" [columns 12-13].
>
> This disclosure teaches that Dutcher does disclose coverage for authentication and control over data/resources relating to the authentication, as well as, maintaining application data in memory until the computer requires/is instructed otherwise.

Appellants respectfully assert that the Examiner is mistaken in the applicability of this portion of Dutcher to the last element of claim 1. The Examiner appears to be implying that the client application discussed in Dutcher at 12:65 – 13:9 [col : line(s)] can read on the claimed "accessible element" of claim 1.

This is not a valid interpretation for a number of reasons. First, although Dutcher does disclose that the client application may be implemented as a set of instructions resident in random access memory (12:65 – 13:1), it does not disclose at what points this program is loaded or unloaded, started or restarted. In fact, Dutcher states, "Until required by the computer, the set of instructions may be stored in another computer memory, for example, in a hard disk drive, or in a removable memory..." (13:1–4) The options discussed would clearly infer a stopping of the computer program and then a subsequent restarting.

But the failure of Dutcher to disclose a subsequent access by a second user to an application or data accessed by a first user goes beyond this fact. The invention disclosed by Dutcher relates to a user authentication mechanism (see title, abstract, disclosure, and claims of Dutcher). Thus, this is what is being referred to in the section cited by the Examiner—i.e., the client application stored in memory is for the preferred implementation of the invention (12:65 – 13:1), i.e., the user authentication program.

But the last element of claim 1 requires that providing access to the accessible element *is dependent on the access rights*—in other words, the second user had to have already been authenticated prior to the access. The client application referred to in Dutcher and cited by the Examiner could not read on the claimed "accessible element" because this client application in Dutcher *is* the authentication element that defines access rights and is accessed by all users equally. Or, stated another way more in line with the claim language, Dutcher discloses *only* an access *independent* on the defined access rights because all users access the authentication routine in the same way and regardless of any authentication or access rights. Furthermore, as noted above, there is no clear teaching in Dutcher as to an unloading or restarting of any application meeting the other requirements of claim 1.

In the AA third paragraph, the Examiner reiterates the presence of multiple users in the system, stating:

> Dutcher discloses "a user of a Windows NT client" [column 6] which implies that there would be one of many users, where each user has access to their own Windows NT client as a part of a domain on a network.
>
> It is noted that any computer system comprising user logon and authentication on a network in a client server configuration comprises at least two or more users, hence the definition of a computer network.

Again, as restated above, it is not sufficient to simply show the presence of multiple users on the system that all have to log in to the system and be authenticated, as is disclosed by Dutcher. Appellants do not dispute the well-known fact that a computer system dealing with authentication issues in a networked computer context strongly infers use by more than one user. However, this is a long way from dealing with the security and data access mechanisms along with the sequencing aspects being claimed here. Even in the discussion of the prior art in the Specification, Applicants note the presence of multiple users and the authentication methods known therefrom. But the present invention requires more than just simply access by a first and second user—it requires that the access be performed without an unloading and reloading (steps currently required in the prior art). The prior art systems are slow because they require, in some way, an unloading or restarting of applications and data by a second user—this is the very problem the present invention deals with.

Claim 1 requires providing access, depending on the defined access rights, for the accessible element that has not been unloaded or restarted, by the second user. The only application disclosed in Dutcher and relied upon by the Examiner is the authentication routine itself, and, for reasons discussed above, this cannot be used to read on the last element of claim 1.

The Examiner has stated many things that are "implied" by the teaching of Dutcher, some of which the Appellants do not disagree with and some that they do disagree with. If the Examiner is taking official notice of the inherency of certain elements related to Dutcher, then this should be expressly stated in the Office Action which would permit the Applicant to challenge such official notice pursuant to MPEP §2144.03.

Since independent claims 10 and 11 comprise subject matter substantially the same as claim 1, they are not argued separately. And since all remaining claims depend from claim 1, Appellants rely on the arguments pertaining to claim 1 as the basis for distinguishing them from the Dutcher reference. It should be noted that the additional reference (Win) does not teach the elements lacking in Dutcher, and was cited by the Examiner for reasons relating to the subject matter added by claim 6.

For the above reasons, Appellants respectfully contend that none of the claims of the present invention are anticipated in view of the disclosure of Dutcher or obviated by a combination of Dutcher and Win.

CONCLUSION

For the above reasons, Appellants respectfully submits that the Examiner is in error in law and in fact in rejecting claims 1–11 based on the teachings of Dutcher. Reversal of the rejection of all of those claims is justified, and the same is respectfully requested.

The previous Appeal Brief was accompanied by an authorization to charge the Applicants' representative's credit card in the amount of $510.00, as required by 37 C.F.R. §41.20(b)(2). It is believed that no additional fee is due for this Resubmitted Brief. However, if necessary, the Commissioner is hereby authorized to charge any additional fees which may be required to or credit any overpayments to account No. 501519.

Respectfully submitted,

/Mark Bergner/ (Reg. No. 45,877)
Mark Bergner
SCHIFF HARDIN, LLP
Patent Department
6600 Sears Tower
233 South Wacker Drive
Chicago, Illinois 60606-6473
(312) 258-5779
Attorneys for Appellant
Customer No. 26574

APPENDIX A
CLAIMS INVOLVED IN THE APPEAL

1. (previously presented) A method for logging a new user into a data processing device with an operating system and an accessible element that is at least one of an application program and sensitive data, comprising the steps of:

ending a first user's access to the accessible element without unloading or restarting the accessible element;

determining authentication data for authenticating a second user;

defining an identity and access rights depending on the authentication data for the second user; and

providing access, depending on the defined access rights, for at [sic: "at" should be removed"] the accessible element, that has not been unloaded or restarted, by the second user.

2. (original) The method according to claim 1, further comprising:

displaying a user interface, depending on the defined access rights;

performing a user switch process step that causes the method to begin again at the first step, content of a user interface remaining unchanged until access rights have been defined again.

3. (original) The method according to claim 2, wherein the content of the user interface is reduced if the renewed definition of access rights defines a more limited scope than the previous definition allowed.

4. (original) The method according to claim 3, further comprising:

generating a warning message indicating a reduction in content and that the user has an opportunity to begin the method at the first step again before the reduction.

5. (original) The method according to claim 1, further comprising:

displaying a user interface in accordance with the access rights that are defined;

deleting, by a User Logout procedure, content of a user interface; and

starting the method from the first step again.

6. (original) The method according to claim 1, further comprising:

logging all access to the application program and all access to the sensitive data together with the respectively defined identity.

7. (original) The method according to claim 1, further comprising:

activating a screen saver by a defined condition to make a user interface illegible; and

beginning the method from the first step again.

8. (original) The method according to claim 7, wherein the defined condition is some amount of elapsed time.

9. (original) The method according to claim 1, further comprising:

blocking all access rights based upon a failed attempt to authenticate a user in the first step.

10. (previously presented) A computer system comprising:

a computer having a memory;

an accessible element that is at least one of an application program and sensitive data that is accessible by a first user and a subsequent second user without unloading or restarting the accessible element;

a program stored in a memory element of the computer memory comprising:

a software module or algorithm for determining authentication data for authenticating the second user with respect to the accessible element;

a software module or algorithm for defining an identity and access rights depending on the authentication data; and

a software module or algorithm for providing access, depending on the defined access rights, for the accessible element.

11. (previously presented) A computer readable data storage media having a program thereon that comprises:

a software module or algorithm for determining authentication data for authenticating a user into a data processing device with an operating system and an accessible element that is at least one of an application program and sensitive data;

a software module or algorithm for defining an identity and access rights depending on the authentication data; and

a software module or algorithm for providing access by the user, depending on the defined access rights, for the accessible element subsequent to an access of the accessible element by a prior first user without unloading or restarting the accessible element.

APPENDIX B
EVIDENCE APPENDIX

There is no additional evidence entered and relied upon for this appeal.

APPENDIX C
RELATED PROCEEDINGS APPENDIX

There are no related proceedings associated with this appeal